

WHAT IS CLAIMED IS:

- 5
1. A method of calling a remote object by a process comprising the steps of:
calling the remote object using a first address in a faulting remote reference to the remote object when the reference refers to an active instance of the remote object; and
calling an activator object using a second address in the faulting remote reference to perform activation of the remote object when the reference to the remote object refers to a null instance of the remote object.
 2. The method of claim 1, further including the step of accessing an interface to call the remote object such that the steps of calling the remote object directly using the first address and calling an activator object using the second address are performed transparently to the process.
 3. The method of claim 1, further including the step of updating the faulting remote reference when a new version of the faulting remote reference is received from a computer associated with the remote object.
 4. A computer readable medium containing instructions executable on a first computer for calling a remote object located at a second computer, the instructions causing the first computer to:
access the remote object directly at the second computer using a first address in a faulting remote reference to the remote object when the reference refers to an active instance of the remote object; and
- 5

08950760-101593

access an activator object at the second computer using a second address in the faulting remote reference to activate the remote object when the reference to the remote object refers to a null instance of the remote object.

5. The computer readable medium of claim 4, further including instructions causing the first computer to access an interface to access the remote object such that the steps of accessing the remote object directly using the first address and accessing an activator object using a second address are performed transparently to a process initiating calling of the remote object.

6. The computer readable medium of claim 4, further including instructions causing the first computer to update the faulting remote reference when a new version of the faulting remote reference is received from the second computer.

7. A method of handling an object call at a remote site for a remote object, the method comprising the steps of:

receiving a call to activate the remote object;

determining whether a first predefined group of objects corresponding to the called remote object is active;

activating the remote object within the first group when the determining step determines that the first group is active; and

creating a second group of objects and activating the remote object within the second group when the determining step determines that the first group is not active.

LAW OFFICES

FINNEGAN, HENDERSON,
FARABOW, GARRETT
& DUNNER, L. L. P.
1300 I STREET, N. W.
WASHINGTON, D. C. 20005
202-408-4000

08950760-101593

8. The method of claim 7, wherein the step of activating the object within the first group further includes the step of activating the object within an address space of previous objects activated within the first group.

9. The method of claim 7, wherein the step of activating the remote object within the first group further includes the step of activating the object within a same Java virtual machine as previous objects activated within the first group.

10. The method of claim 7, wherein the step of creating the second group includes spawning a virtual machine to interpret the second group.

11. The method of claim 10, further including the step of returning results of the activated remote object.

12. A computer readable medium containing instructions executable on a remote computer in a network of distributed computers for handling an object call at the remote computer for a remote object, the instructions causing the computer to:

determine whether a first predefined group of objects corresponding to the called remote object is active;

activating the remote object within the first group when the determining step determines that the first group is active; and

creating a second group and activating the remote object within the second group when the determining step determines that the first group is not active.

13. The computer readable medium of claim 12, wherein the step of activating the remote object within the first group further includes further including instructions for performing

the step of activating the object within an address space of previous objects activated within the first group.

14. The computer readable medium of claim 12, wherein the step of activating the remote object within the first group further includes instructions for performing the step of activating the object within a same Java virtual machine as previous objects activated within the first group.

15. The computer readable medium of claim 12, wherein the step of creating the second group further includes instructions for spawning a virtual machine to interpret the second group.

16. The computer readable medium of claim 12, further including instructions for performing the step of returning results of the activated remote object.

17. A distributed computer network comprising:

a first computer executing a proxy object called by a process and instantiated as one of a plurality of different implementations depending on the process call; and

a second computer receiving requests for remote objects from the first computer and executing an object activator performing the steps of: (1) determining whether a first predefined group of objects corresponding to the requested remote object is active; (2) activating the requested remote object within the first group of objects when the determining step determines that the first group of objects is active; and (3) creating a second group of objects and activating the requested remote object within the second group of objects when the determining step determines that the first group of objects is not active.

08950760-101593

0850760-101597

18. The computer system of claim 17, wherein the plurality of different implementations of the proxy object form an interface such that details of the remote object call are hidden from the calling process.

19. The computer system of claim 17, wherein the proxy object further includes means for calling the remote object directly using a first address in a faulting remote reference to the remote object when the reference refers to an active instance of the remote object and means for calling the activator object using a second address in the faulting remote reference when the reference refers to a null instance of the remote object.

20. The computer system of claim 17, wherein the proxy object is a processes executing on a virtual machine.

21. The computer system of claim 17, wherein the object activator is a processes executing on a virtual machine.

22. A method of calling a remote object comprising the steps of:
receiving a request to access the remote object;
determining whether the remote object is active; and
accessing the remote object based on the results of the determination.

23. The method of claim 22, wherein the determining step includes the substep of maintaining a faulting remote reference to the remote object.

24. The method of claim 22, wherein the determining step includes the substep of maintaining a faulting remote reference to the remote object, and when the faulting remote

LAW OFFICES

FINNEGAN, HENDERSON,
FARABOW, GARRETT
& DUNNER, L. L. P.
1300 I STREET, N. W.
WASHINGTON, D. C. 20005
202-408-4000

reference indicates that the remote object is active, the accessing step includes the substep of directly contacting the remote object.

25. The method of claim 22, wherein the determining step includes the substep of maintaining a faulting remote reference to the remote object, and when the faulting remote reference indicates that the remote object is not active, the accessing step includes the substep of instantiating the remote object.

26. The method of claim 25, wherein the instantiating step includes the substep of determining whether a virtual machine for the remote object is active.

27. The method of claim 26, wherein when the a virtual machine for the remote object is not active, the instantiating step further includes the step of spawning a new virtual machine for the remote object.

ADD AL
B
EI

08950760-101597

LAW OFFICES

FINNEGAN, HENDERSON,
FARABOW, GARRETT
& DUNNER, L.L.P.
1300 I STREET, N.W.
WASHINGTON, D. C. 20005
202-408-4000